Application note for CMV2000 and CMV4000

Interface Training

Change record

Issue	Date	Modification
v1	09/12/09	Origination

Table of Contents

1 INTRODUCTION

The CMV2000/CMV4000 sensors have an 18-channel LVDS interface, consisting of:

- 1 output DDR clock channel
- 1 control channel to transfer status information
- 16 data channels to transfer the 10 or 12 bit pixel data

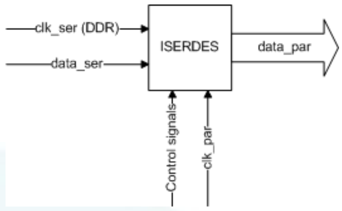
To ensure a correct and stable interface between sensor and FPGA, a training routine should be run before grabbing images. This application note describes the training algorithm that is used in CMOSIS characterization and evaluation systems.

In the CMOSIS systems, a Virtex 4 FPGA is used. Some primitives and concepts used and explained in this document apply to this FPGA and might be different for other FPGA types and vendors. It's beyond the scope of this document to go into detail on the implementation an all popular or available platforms.

This document is accompanied by the VHDL implementation of the receiver.

2 Architecture

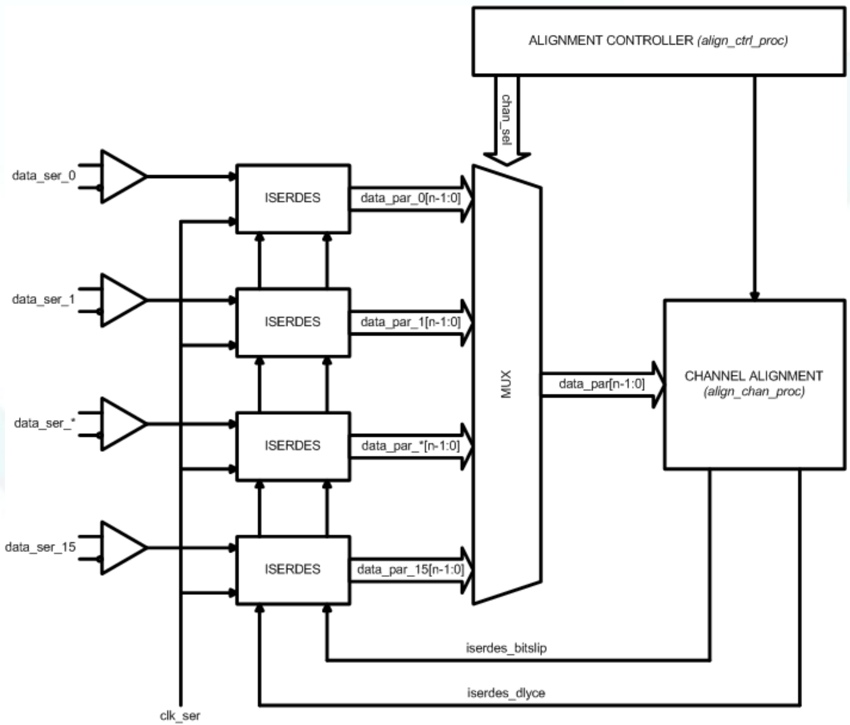
Deserialization is done with an ISERDES module for every channel. It converts the serial data stream to parallel data. A number of control signals are used to control the behavior of the ISERDES module.



The goal of the training is bit and word alignment of all LVDS channels.

- Bit alignment
  - is done to ensure that that a channel is sampled in the center of its eye diagram.
  - is achieved by adding delay to the input data path of the channel (using the variable IOBDELAY functionality of the ISERDES).
- Word alignment
  - is done to ensure that bit 0 of all channels is sampled at the same clock edge
  - is achieved by rotating the received parallel word until the parallel output matches the training word (using the BITSLIP functionality of the ISERDES).

Next figure shows the architecture of the receiver in the FPGA.



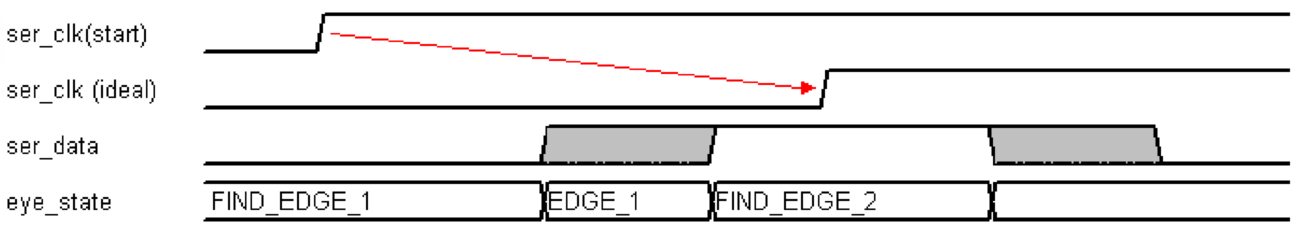
To save resources, the alignment controller is shared by all channels. The parallel data from all channels is multiplexed to the channel alignment block where the n channels are aligned one by one. The alignment controller block sequences the events in the receiver block.

3 Channel alignment

3.1 Bit alignment

The goal of the bit alignment is to place the sampling point in the ideal position for each channel. This is done by adding delay to the data input so that it moves relative to the sampling clock.

One bit period consists of two regions: a stable and an unstable region. Sampling in the stable region guarantees that the correct data will always be sampled, regardless of the number of samples that are acquired. In the unstable region, the chance of sampling correct data is not 100%. The unstable region exists because of the non-ideal rise and fall times between two bits and because of jitter on data and clock. Next figure shows one bit period with the unstable region in grey. The ideal sampling position is in the middle of the stable region. Finding this position is the goal of the bit alignment.



Determining if the sampling point is in a stable or unstable region is done by making *N* samples of the bit. If the sampled data has *N* times the same value, the sampling point is considered to be stable. The number of samples *N* should be high enough for a decent statistical coverage.

Finding the start and end of the bit period is done by continuously determining if a selected sampling point is stable or unstable, adding delay and checking again. At the start of the bit alignment routine, the relative position of clock and data is not known. Therefore, the routine will shift the sampling position until it finds an unstable point ("FIND_EDGE_1"). At this moment, the controller knows that it is sampling near a transition between two bits. From this point on, the controller will continue shifting the sampling point in the same direction until it finds a stable sampling point ("EDGE_1"). Now, the controller has found the start of the stable region. The only thing that remains now is finding the end of the stable region. This is done by shifting the sampling point until an unstable sampling point is found ("FIND_EDGE_2"). If this point is found, the controller knows the boundaries of the stable bit period and it can place the sampling point in the middle.

3.2 Word Alignment

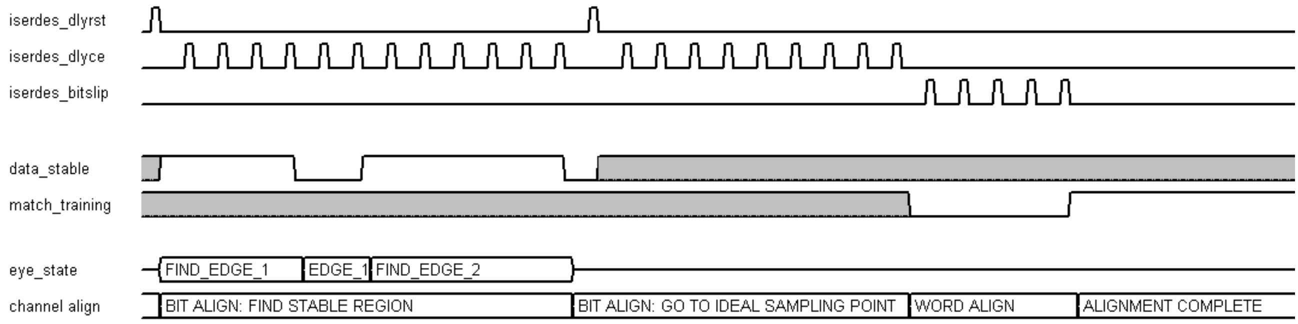
The serial data channel contains a continuous stream of bits. It is impossible for the receiver to know the position of the first bit of a 10/12-bit word. The goal of the word alignment is finding the position of the first bit of a word.

Word alignment is done by continuously sampling the 10/12-bit training word that the sensor transmits when it is idle. The value of the training word on the data channels can be set via register upload (default = 85). If the sampled 10/12-bit word does not match the expected training word, the sampling point of the first bit of a word is moved by 1 bit period. This is done until the training word matched.

(Note that the training word on the control channel is not programmable via register setting, but fixed at 512.

3.3 Timing example

Next figure shows an example of the channel alignment.



The ISERDES is controlled with three control signals:

- iserdes_dlyrst reset the delay of the input path of the ISERDES module to zero.
- iserdes_dlyce increases the delay of the input path of the ISERDES
- iserdes_bitslip moves the sampling point of the first bit by one bit period

The channel alignment starts with the bit alignment. The first stage of the bit alignment is FIND_EDGE_1. In the example, the delay is increased 4 times before the first unstable sampling point is found (data_stable is '0'). After two more delay steps, the sampling point stable again. This means the start of the stable region is after 6 delay steps. After 6 more delay steps, the data is no longer stable. The stable region is between 6 and 12 delay steps, so the ideal sampling position is 9 delay steps (=> (6+12)/2).

If the ideal sampling position is found, the input delay is reset and the sampling point is moved to the ideal location. If this is done, the bit alignment for the channel is completed. The data word is checked to see if it matches the training word. If it is not the case, the bitslip signal is asserted to move the word position. In the example, after 5 bitslips, the received data matches the expected training word and the training of the channel training is finished.

3.4 STATE DIAGRAM

Next figure shows the state diagram of the channel alignment controller. After correct execution of these states, 1 channel is correctly aligned.